Towards Open Data Sets (k)-Anonymization

of Patient EHR Data

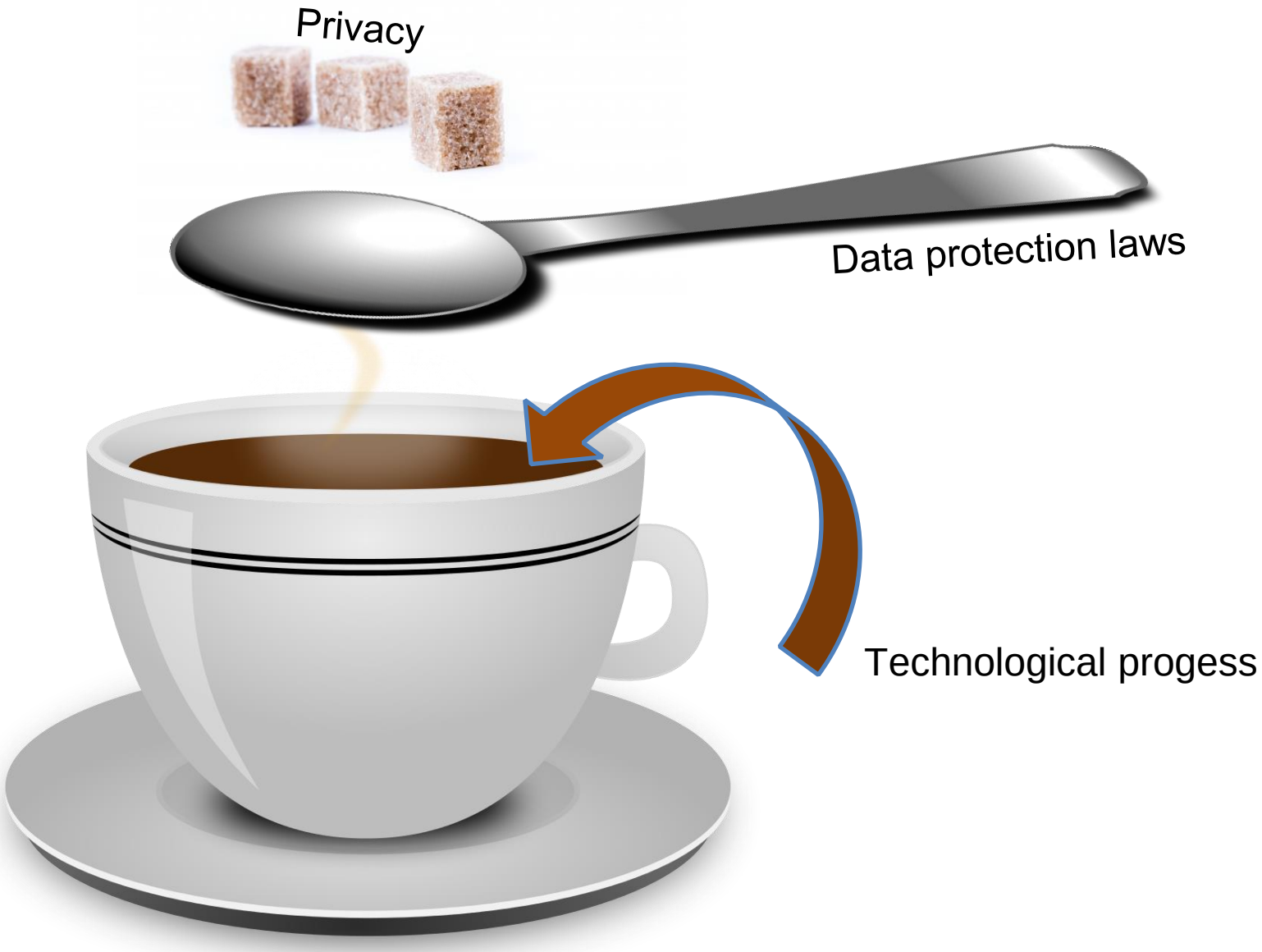
Bernd Malle
b.malle@hci-kdd.org

Holzinger Group - www.hci-kdd.org

Overview / Contents

1. Introduction & Motivation
   - The right to be forgotten & it's consequences
2. Perturbation vs. Anonymization
3. ML performance on perturbed data
4. K-Anonymization - Motivation - Basics
5. Limits of anonymization (k, l, t)
6. (Some) Algorithmic Approaches
   - Greedy clustering
   - SaNGreeA
7. ML performance on anonymized data
8. Can iML help in anonymization?

Privacy in the 21st century… ??



The right to be forgotten

- Basically: A user has the right to have their data deleted from a database upon request
- In past cases, the requirement only meant deletion from a search index (due to EU tech ignorance)
- From 2018 onwards, the “right to be forgotten” will be part of the new EU data protection rules
- Since one cannot foresee which (non-existing) laws will be enforced by the European bureaucracy in the future (see Apple..), it would be wise to be prepared…
- There is even a proposal by German data protection advocates to restrict automated processing of anonymized data which “might” be de-anonymizable…

Impact on different DB layers

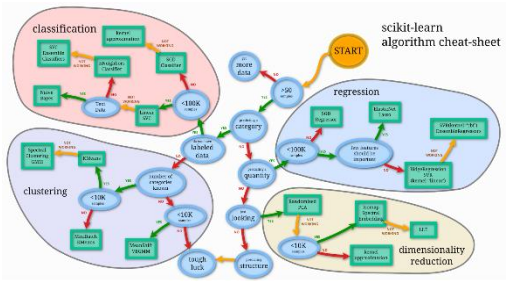
OLTP

Less convenient service?

Backups

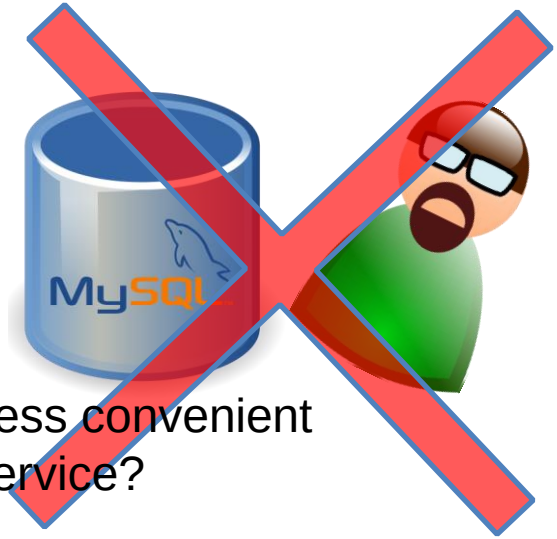
System failures?

Incomplete statistics?



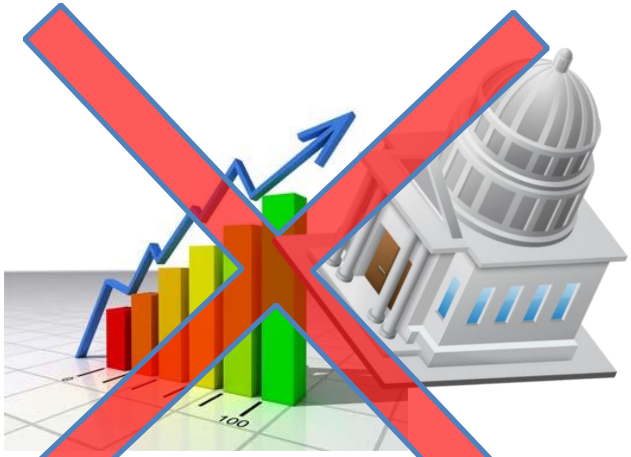
ML performance…?

Whom does it concern?

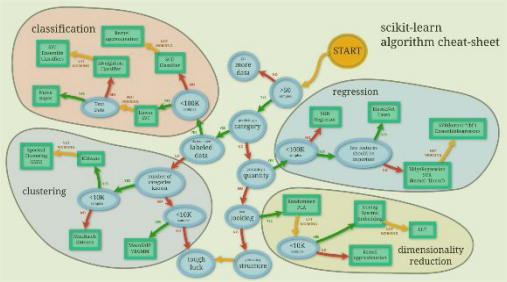


Less convenient service?

Re-appearing items?



Incomplete statistics?



ML performance…?

Two different experimental scenarios

1. Simulate users exercising their “right to be forgotten” in the worst way possible – requesting the erasure of the most valuable data points in the knowledge base.

In the future extendable to

- Outlier deletion first (anomalous users have higher probability to request their data deleted)
- Perturbation via addition of ‘targeted’ noise

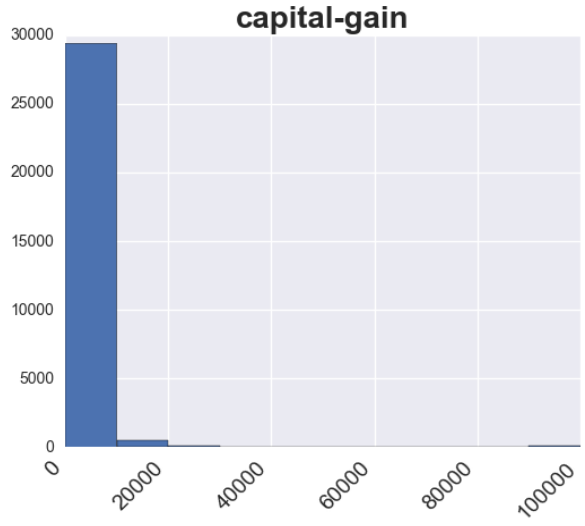
2. Try to circumnavigate the re-creation of our ML databases by anonymizing them in the first place and applying our learning algorithms on that anonymized datasets.

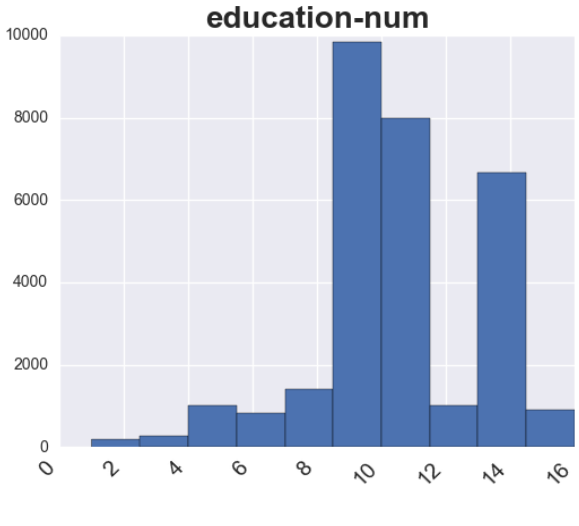
Scenario One

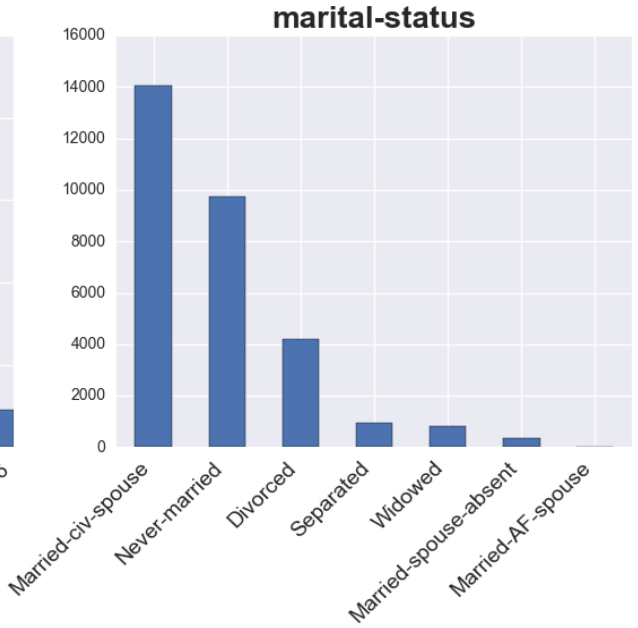
implies

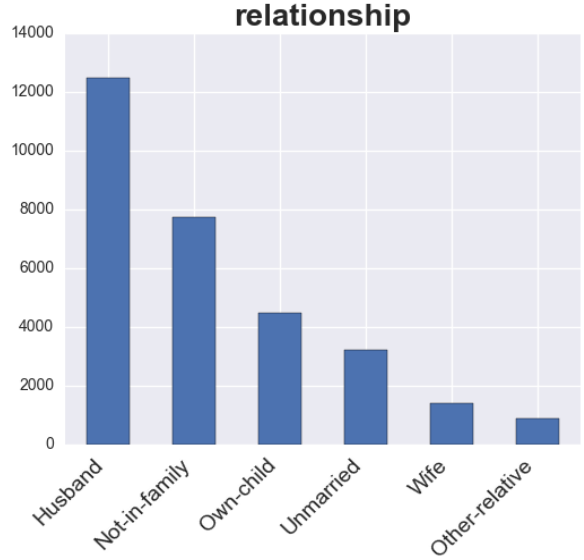
Selectively deleting (valuable) data points

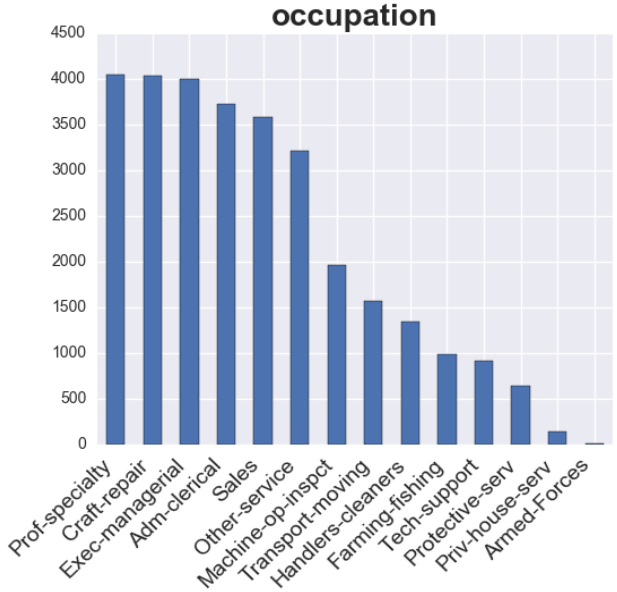
Adult dataset original distribution

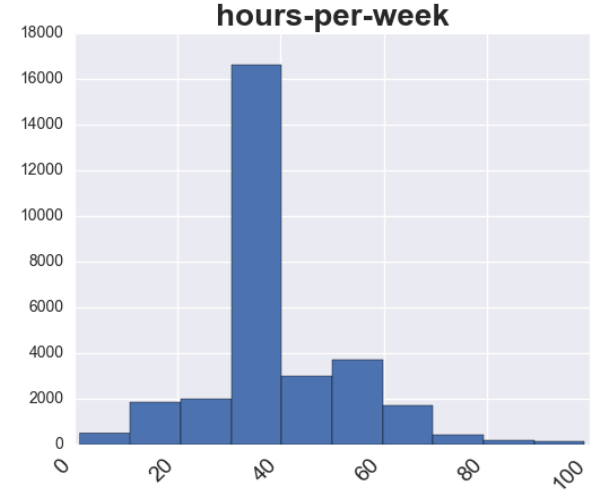






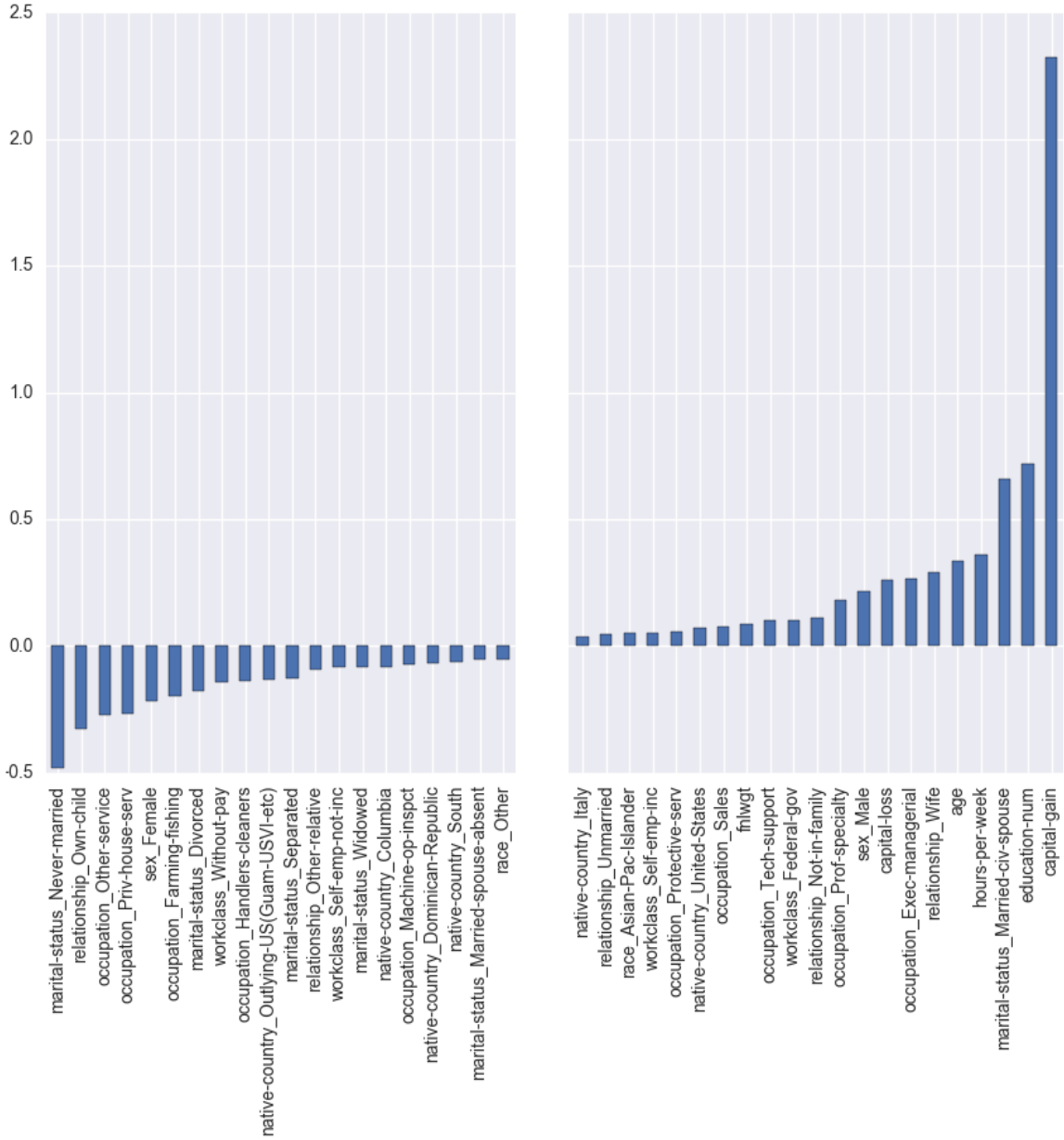






Find the most valuable data points

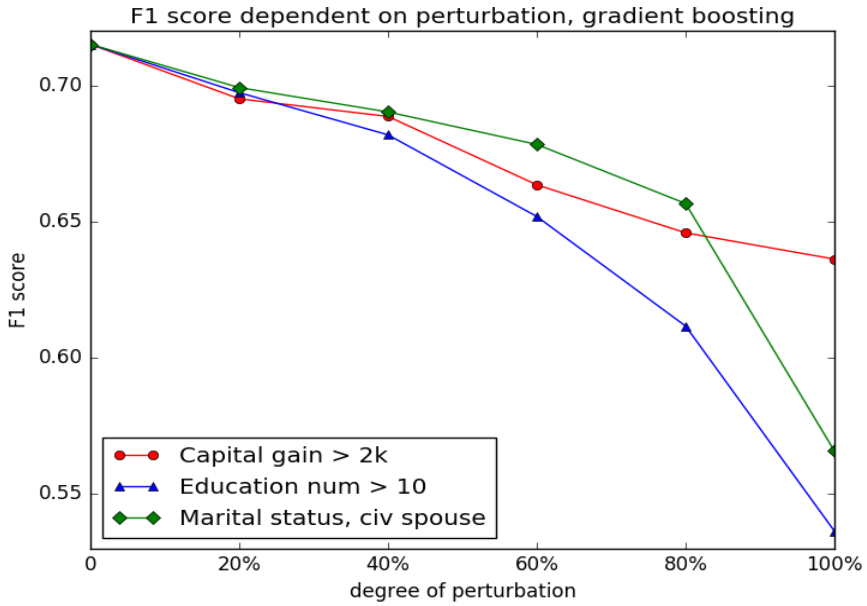
1. Preprocess dataset
2. Train some logistic classifier on it
3. Retrieve the coefficients learned by the Log.Class.
4. Sort & use the best xyz as most valuable columns

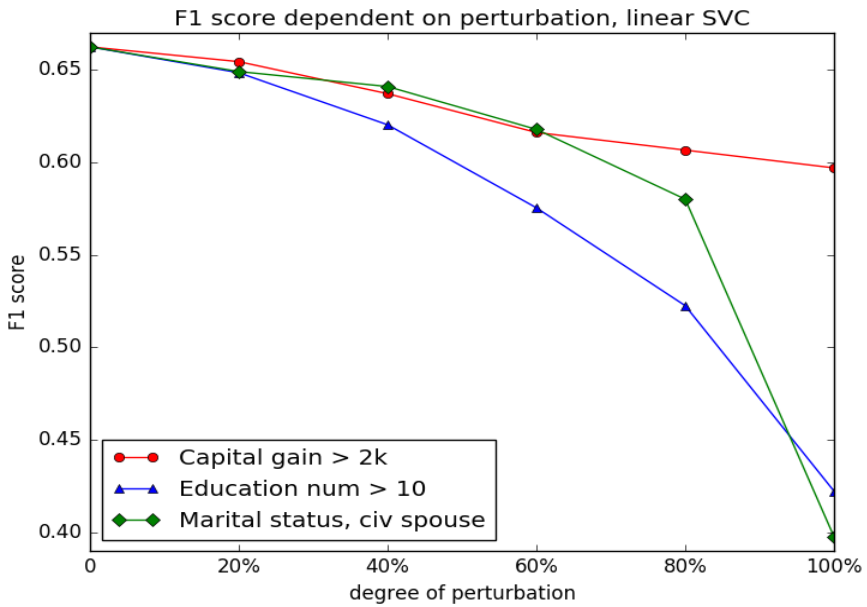


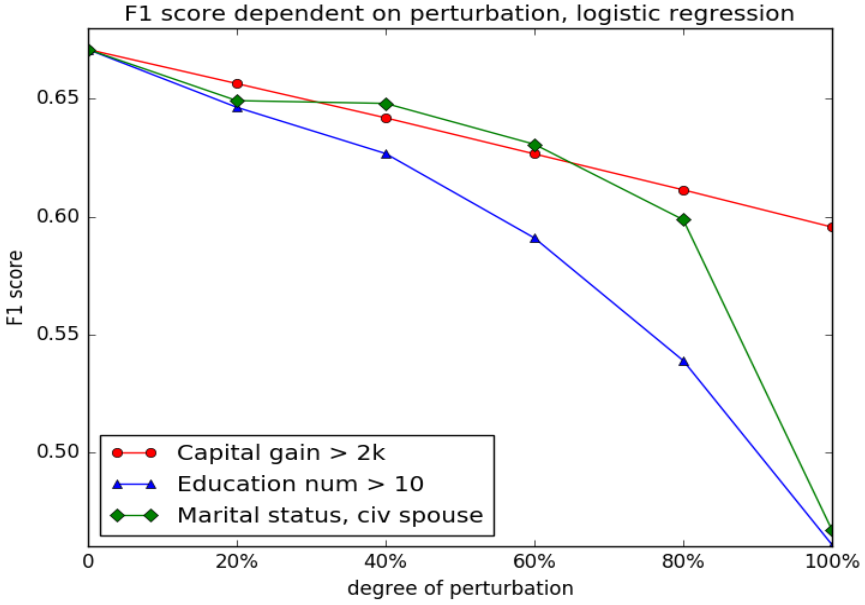
Selective deletion – prepared datasets

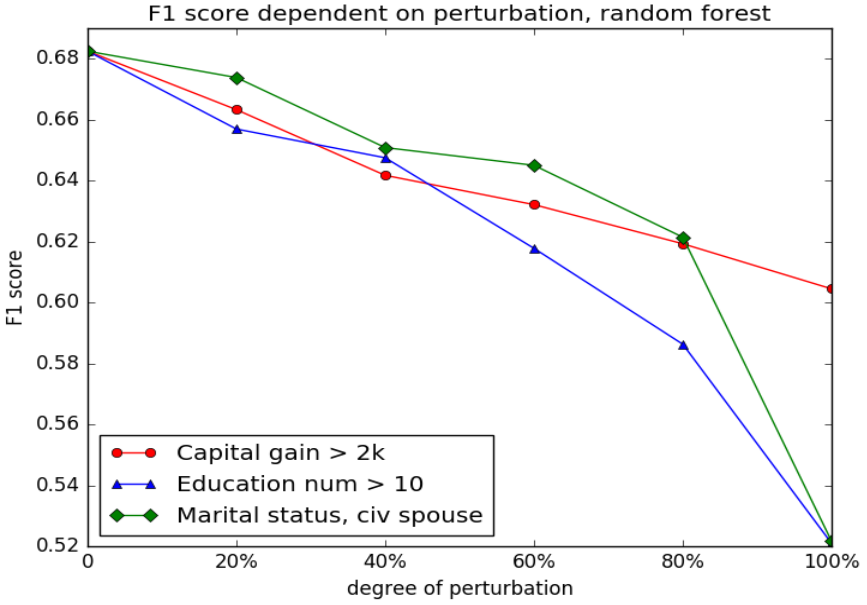
- After extracting the 3 attribute values contributing the most information to the classifier
- We construct new datasets with 0.2, 0.4, 0.6, 0.8 and 1.0 fractions of those data rows missing
- Thereby constructing 15 new data sets
- To use 4 different classifiers on…

Selective deletion - Results









Scenario Two

implies

Wholesale anonymization of the knowledge base

- Public release of sensitive information is useful for
  - Statistics => education, grant proposals ;-)
  - Research => prediction of disease spreading etc.

- However, personal identities need to be concealed

- In the past, simple approaches have failed to provide sufficient security:
  - data linkage of publicly available datasets
    - Netflix database, which was linked with the IMDB movie ratings database (via date of rating) => at least one user was re-identified

Re-Identifying the NYC Taxi Ride Dataset

1. Find suspicious data
2. Figure out what ONE hash represents (‘0’)
3. Figure out input domain for hashes
   => Medallions are 4-5 digits
   => ~20M possibilities
4. Construct inverted LUT
5. DS hacked !!!

We need robust anonymization techniques

Data properties => Reduce granularity

Name	Age	Zip	Gender	Disease
Alex	25	41076	Male	Allergies
…	…	…	…	…

- Identifiers := immediately reveal identity
  - name, email, phone nr., SSN

  => DELETE
- Sensitive data
  - medical diagnosis, symptoms, drug intake, income

  => NECESSARY, KEEP
- Quasi-Identifiers := used in combination to retrieve identity
  - Age, zip, gender, race, profession, education

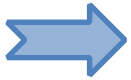
  => MAYBE USEFUL

  => MANIPULATE / GENERALIZE

Anonymization criteria 1/4

k-anonymity: for every entry in the DS, there must be at least k-1 identical entries (w.r.t. QI's) => this is 3-anon:

Node	Name	Age	Zip	Gender	Disease
X1	Alex	25	41076	Male	Allergies
X2	Bob	25	41075	Male	Allergies
X3	Charlie	27	41076	Male	Allergies
X4	Dave	32	41099	Male	Diabetes
X5	Eva	27	41074	Female	Flu
X6	Dana	36	41099	Female	Gastritis
X7	George	30	41099	Male	Brain Tumor
X8	Lucas	28	41099	Male	Lung Cancer
X9	Laura	33	41075	Female	Alzheimer



Node	Age	Zip	Gender	Disease
X1	25-27	4107*	Male	Allergies
X2	25-27	4107*	Male	Allergies
X3	25-27	4107*	Male	Allergies
X4	30-36	41099	*	Diabetes
X5	27-33	410**	*	Flu
X6	30-36	41099	*	Gastritis
X7	30-36	41099	*	Brain Tumor
X8	27-33	410**	*	Lung Cancer
X9	27-33	410**	*	Alzheimer

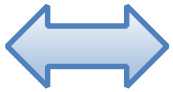
There are 2 possible attacks on k-anonymity though…

Trade-off between:

- Data utility => min. information loss
- Privacy => max. information loss

Both can be easily achieved (but not together ☺)

Node	Name	Age	Zip	Gender	Disease
X1	Alex	25	41076	Male	Allergies
X2	Bob	25	41075	Male	Allergies
X3	Charlie	27	41076	Male	Allergies
X4	Dave	32	41099	Male	Diabetes
X5	Eva	27	41074	Female	Flu
X6	Dana	36	41099	Female	Gastritis
X7	George	30	41099	Male	Brain Tumor
X8	Lucas	28	41099	Male	Lung Cancer
X9	Laura	33	41075	Female	Alzheimer



Node	Age	Zip	Gender	Disease
X1	*	*	*	Allergies
X2	*	*	*	Allergies
X3	*	*	*	Allergies
X4	*	*	*	Diabetes
X5	*	*	*	Flu
X6	*	*	*	Gastritis
X7	*	*	*	Brain Tumor
X8	*	*	*	Lung Cancer
X9	*	*	*	Alzheimer

1. Homogeneity attack:
   - all entries contain the same piece of sensitive information (Allergies)

Node	Age	Zip	Gender	Disease
X1	25-27	4107*	Male	**Allergies**
X2	25-27	4107*	Male	**Allergies**
X3	25-27	4107*	Male	**Allergies**

2. Background knowledge attack:
   - Given two entries with identical QI sets: One has lung cancer, the other diabetes…

Node	Age	Zip	Gender	Disease
X8	27-33	410**	*	Lung Cancer
X9	27-33	410**	*	Diabetes

l-diversity: for every "equivalence class" of (at least k) QI-duplicates, there must be at least *l* different "well represented" values for the sensitive attribute

2 possible attacks:

1. Skewness attack:
   - cancer = positive 1% / negative 99%
   - Chances are still
2. Semantic closeness attack:
   - gastritis / gastric ulcer

Node	QI	Cancer	Drugs
X1	*	Y	xyz…
X2	*	Y	xyz…
X3	*	Y	xyz…
X4	*	N	xyz…
X5	*	Y	xyz…
X6	*	Y	xyz…
X7	*	N	xyz…
X8	*	N	xyz…
X9	*	N	xyz…

t-closeness: an equivalence class has t-closeness if the intra-class distribution of a sensitive attribute differs no more than a threshold t from it's global distribution (whole dataset). The whole DS has t-closeness if this holds for every equivalence class it contains.

basic idea:

- we do not want an attacker to gain too much insight (additional information) by looking at the data
- additional information => surprise (delta expectation)
- the closer our local and global distributions are => the less our local group deviates from expectations

Different kinds of data input format

1. Microdata
   - data at the granularity of individuals (table row)

2. Graph data -> social network data, in which
   - nodes represent microdata
   - edges represent their structural context
   - graph data are harder to anonymize
     - It's harder to model the background knowledge of an attacker.
     - It is harder to quantify the information loss of modifications.
     - Modifications can propagate through the network.

Perturbative

- Adding noise only distribution counts
  - Value perturbation => numerical attributes
  - Graph perturbation
    - (randomly) adding / deleting nodes / edges

- Microaggregation / Clustering
  - Replace node data by centroid data
  - good for numerical data, but possible also for others given rules
  - Ensures k-anonymity only when computed over all attributes at the same time
  - Exact optimal only in P when computed over just 1 attribute (else heuristic)

Non-perturbative

- Generalization (hierarchies)
  - fixed ruleset
  - range partitioning (numerical values...)

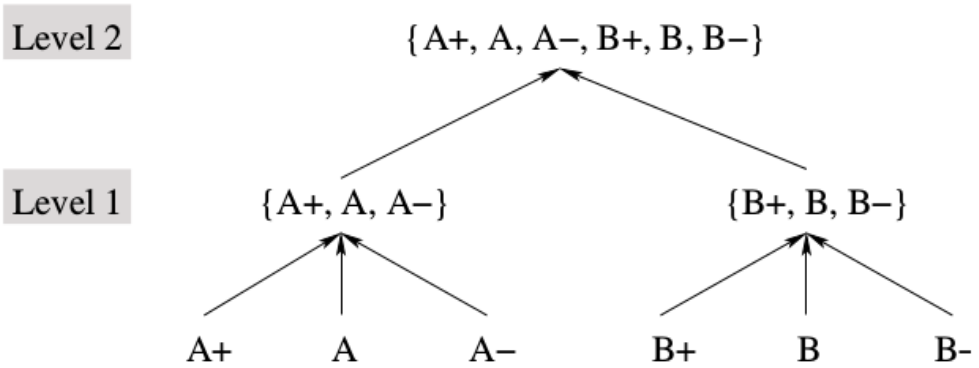


Figure 1: A possible generalization hierarchy for the attribute "Quality".

- Suppression
  - Special case of generalization (with one level)

Graphics Source: Bayardo, R. J., & Agrawal, R. (2005, April). Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on* (pp. 217-228). IEEE.

“Social Network Greedy Anonymization” (SaNGreeA)

- Anonymizes a dataset w.r.t 2 information categories:
  - Feature vector values => traditional, tabular
  - Graph structure => edge configuration

- Based on the concept of ‘greedy’ clustering

- Which poses the question:
  - How do we choose the next node to add to a cluster w.r.t the above two criteria?

! We need some (good) cost functions !

Greedy anonymization Main Loop

```
## MAIN LOOP
for node in adults:
    if node in added and added[node] == True:
        continue
    # Initialize new cluster with given node
    cluster = CL.NodeCluster(node, adults, adj_list, gen_hierarchies)
    # Mark node as added
    added[node] = True
    # SaNGreeA inner loop - Find nodes that minimize costs and
    # add them to the cluster since cluster_size reaches k
    while len(cluster.getNodes()) < GLOB.K_FACTOR:
        best_cost = float('inf')
        for candidate, v in ((k, v) for (k, v) in adults.items() if k > node):
            if candidate in added and added[candidate] == True:
                continue
            cost = cluster.computeNodeCost(candidate)
            if cost < best_cost:
                best_cost = cost
                best_candidate = candidate
        cluster.addNode(best_candidate)
        added[best_candidate] = True
    # We have filled our cluster with k entries, push it to clusters
    clusters.append(cluster)
```

- Generalization Information loss (GIL)
  - Based on content of nodes

- We assume
  - Continuous properties (age, body height, …)
    - Candidate Nodes hold a particular value
    - Clusters have either particular value (at the start) or a generalized range
    - In order to incorporate the node into the cluster, we may have to generalize this range further, increasing the cost.

  - Categorical properties (work class, native-country, …)
    - Same preconditions as above
    - We use generalization hierarchies to determine the cost of clustering

- Generalization information loss function:

$$GIL(cl) = |cl| \cdot \left(\sum_{j=1}^{s} \frac{size(gen(cl)[N_j])}{size(min_{X \in \mathcal{N}}(X[N_j]), max_{X \in \mathcal{N}}(X[N_j]))} + \sum_{j=1}^{t} \frac{height(\Lambda(gen(cl)[C_j]))}{height(\mathcal{H}_{C_j})}\right),$$

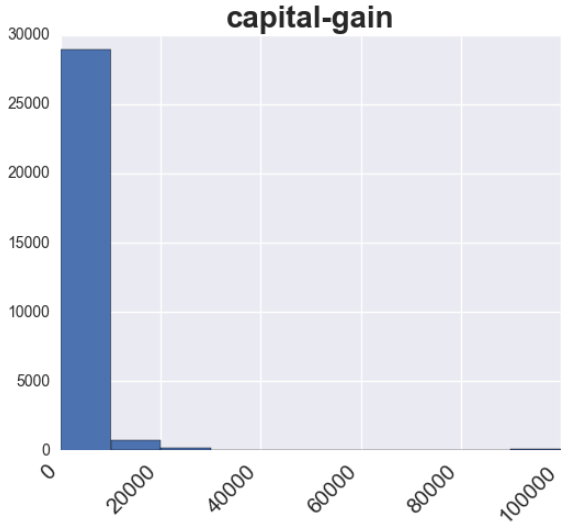
where:

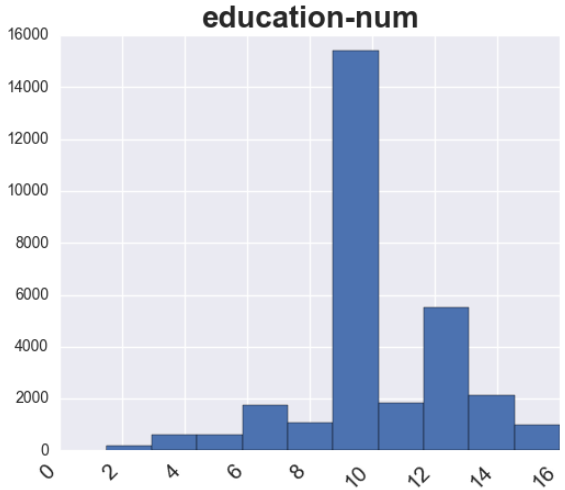
- $|cl|$ denotes the cluster cl's cardinality;
- $size([i_1, i_2])$ is the size of the interval $[i_1, i_2]$, i.e., $(i_2 - i_1)$;
- $\Lambda(w)$, $w \in \mathcal{H}_{C_j}$ is the subhierarchy of $\mathcal{H}_{C_j}$ rooted in w;
- $height(\mathcal{H}_{C_j})$ denotes the height of the tree hierarchy $\mathcal{H}_{C_j}$.

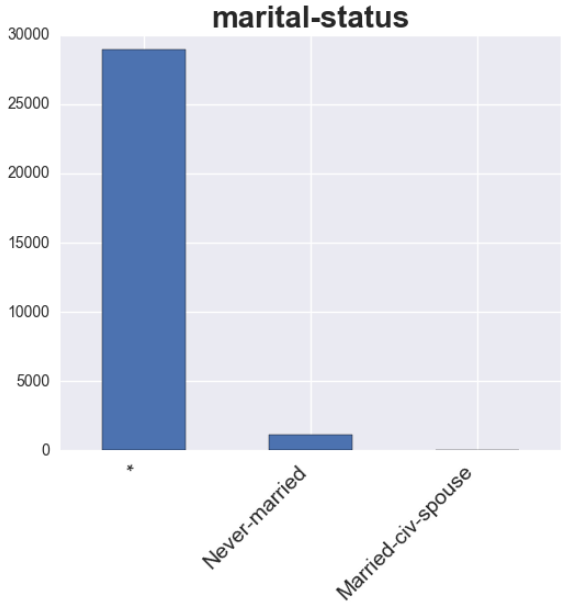
Campan, A. and Truta, T.M., 2009. Data and structural k-anonymity in social networks. In *Privacy, Security, and Trust in KDD* (pp. 33-54). Springer Berlin Heidelberg.

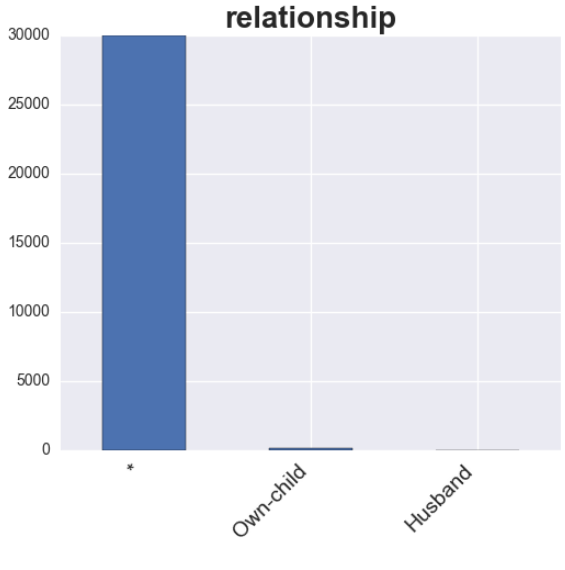
- Example GIL:
  - age_range overall = [11 – 91]
  - In order to cluster some nodes, we need to generalize 27 to [20 - 30]
  - Cost = (30-20)/(91-11) = 1/8
  - Given a generalization hierarchy ‘native-country’ with 4 levels
  - In order to cluster, we need to generalize ‘Austria’, ‘France’, or ‘Portugal’ to ‘Western Europe’, which is 1 level higher
  - Cost = 1/4

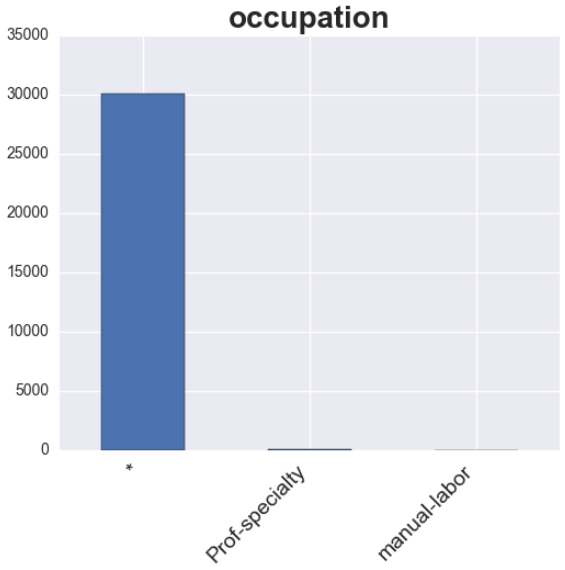
Anonymization - datasets

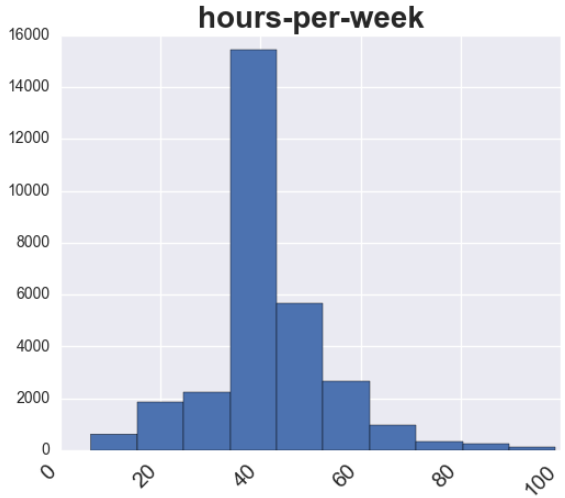








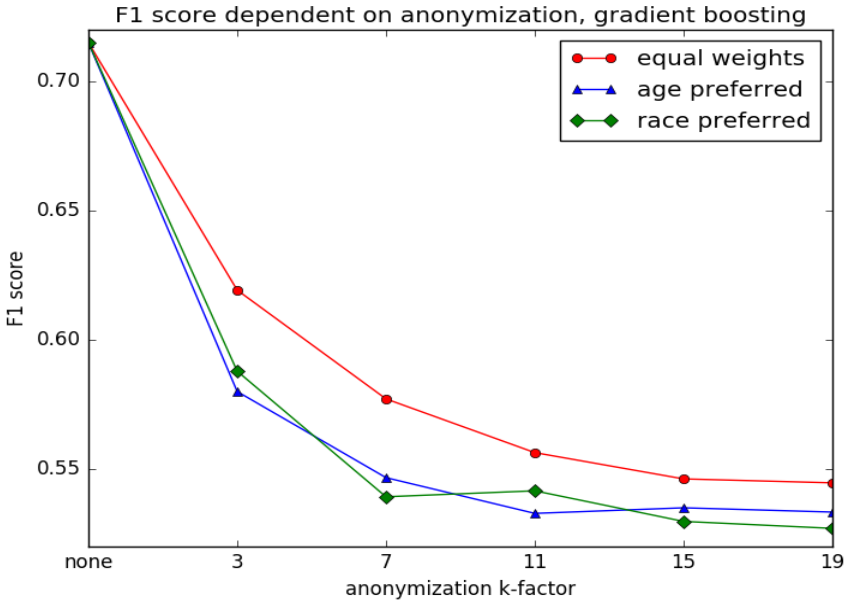


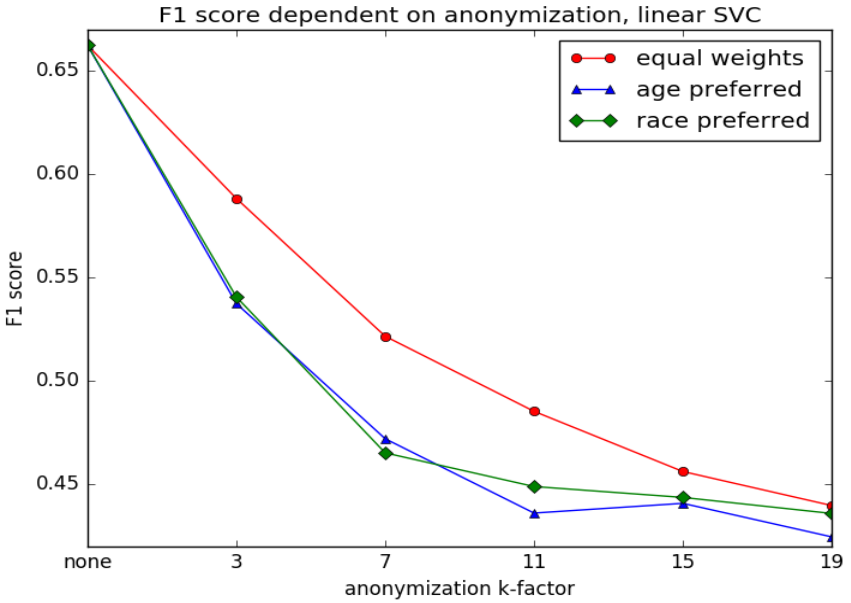


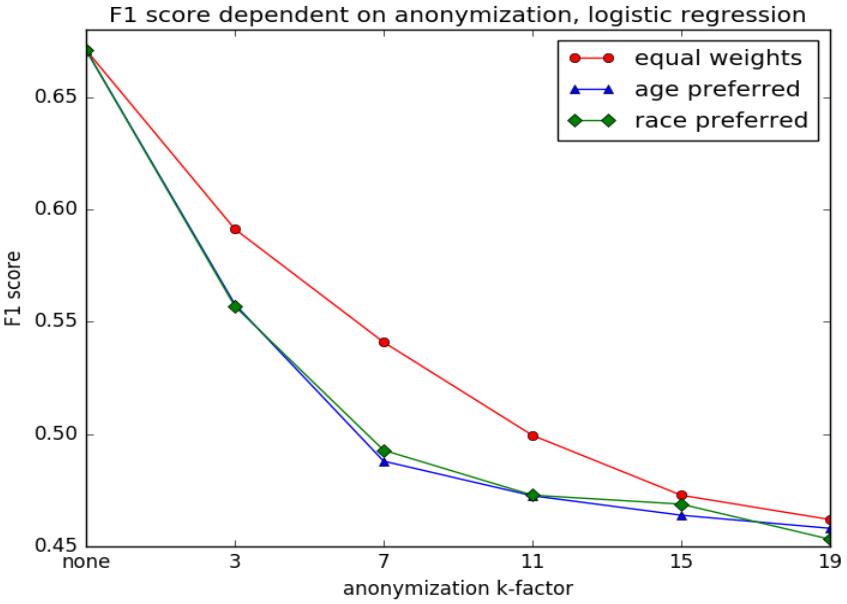
Anonymization – prepared datasets

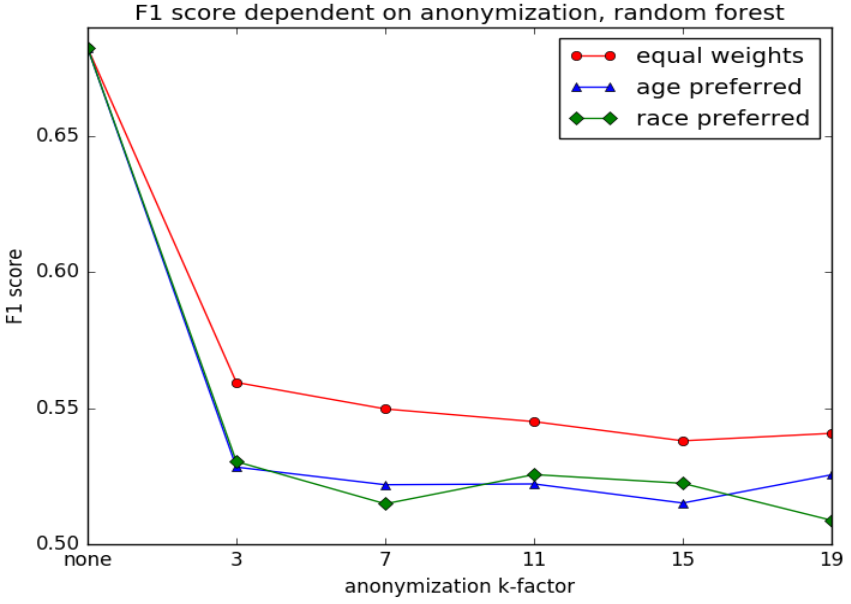
- We used k-factors of:
- 3, 7, 11, 15 and 19
- Each combined with three different weight vectors
  - Equal weights for all columns
  - Age preferred (0.88 vs 0.01 rest)
  - Race preferred (0.88 vs. 0.01 rest)
- Resulting in 15 differently anonymized data sets

ML on Anonymization - Results









Bernd Malle <b.malle@hci-kdd.org>

Initial Conclusions

1. Succumbing to the “right-to-be-forgotten” still seems better than performing ML on anonymized DBs

2. A whole lot of future research is needed in order to corroborate and expand on those results

   - Extension to other ML approaches
     => Prediction, Clustering, Dim. Reduction, Pattern Rec.
   - Other perturbation techniques
   - Graph-based datasets

Can iML help anonymization 1/4 ?

Examples of iML?

- The CAT (Cornell anonymization toolkit) as well as ARX (TU Munich) allow you to run utility / risk analysis
- However, they are not interactive, but only support re-running your experiment with new settings…

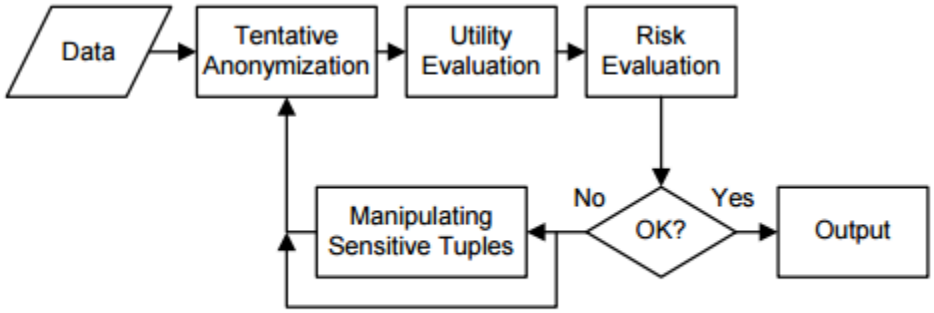
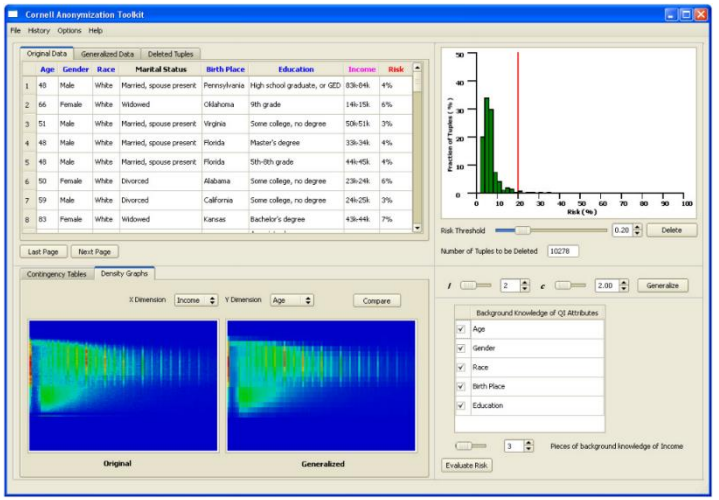


Figure 2: Anonymization process



Possibilities to bring iML into anonymization?

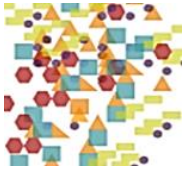
1. Distance functions for Clustering
2. Information loss measures

- Both are subjective
- “Optimality” will also depend on the specific domain (medical vs. financial data)
- So (inter)active learning could be applied by involving a domain expert => the Human-in-the-loop approach…

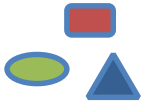
Can iML help anonymization 3/4 ?

Can iML help anonymization 4/4 ?

Case: data similarity:



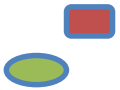
Subset of Data



Sample presented to User

Which two are more similar?

User decides



Update data + learn Heuristics

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

Weight Vectors 1/2

[55 - 76]	*	North_America	Male	*	Married-civ-spouse
[55 - 76]	*	North_America	Male	*	Married-civ-spouse
[55 - 76]	*	North_America	Male	*	Married-civ-spouse

51 | Private | United-States | Male | White | Married-civ-spouse

[48 - 70]	Private	America	Male	White	*
[48 - 70]	Private	America	Male	White	*
[48 - 70]	Private	America	Male	White	*

Weight Vectors 2/2

Applying a weight vector to our desired columns wi ll change our cost function and thereby produce dif ferent anonymization results:

age	workclass	native-country	sex	race	marital-status
0.1667	0.1667	0.1667	0.1667	0.1667	0.1667

age	workclass	native-country	sex	race	marital-status
0.95	0.01	0.01	0.01	0.01	0.01

Conclusion

- Conclusion: the level of privacy / security of data will always remain subjective with regard to the data set as well as potential attackers !!
- You can never answer the question: "Will this algorithm be good enough for our purposes?" without testing it thoroughly for your specific use cases on YOUR OWN DATA…
- Data that might seem safe today might become unsafe again in the future (additional

Thank you!